UDC 044.9. 629.7.05 DOI: 10.18372/2073-4751.70.16846

Kovalenko Yu.B., Candidate of Sciences in Pedagogy,
orcid.org/0000-0002-6714-4258,
Kudrenko S.O., Candidate of Sciences in Technology,
orcid.org/0000-0002-0759-3908

METHODOLOGY FOR TESTING LANGUAGES FOR EMBEDDED AVIONICS SYSTEMS

National Aviation University

yleejulee22@gmail.com
stanislava@i.ua

Introduction

Specificities of In-the-Loop Testing. Figure 1 provides a schematic view of the life cycle of an avionic embedded system. After the system specification and design phases, software and hardware components are developed and individually tested before software and hardware integration testing proceeds. At the end of the development process, the target system – together with additional avionic embedded systems and with hydraulic, mechanical, and other systems – is embedded into an aircraft prototype (Iron Bird). Later, a flight test program is performed. Once certification has been passed, production and maintenance processes are entered. The application logic does not need further functional validation, but hardware testing activities are still necessary to reveal manufacturing and aging problems.

This paper focuses on in-the-loop testing phases that occur during the development process. We introduce below the specificities of in-the-loop testing, as they have an impact on the languages analyzed in this paper [5-9].

Materials and methods

The Various Forms of In-the-Loop Testing:

- An avionic system is tightly coupled to its environment. Testing the functional logic requires producing a large volume of data that, in the operational environment, would come from other avionic systems and physical sensors.
- In-the-loop testing addresses this problem by having a model of the environment to produce the data. The model of the environment receives the outputs of the system under test (e.g., commands to actuators) and computes the next inputs accordingly. In the case of a control system, computation must typically account for the physical rules governing the dynamics of the controlled aircraft elements.

As shown in Figure 1, in-the-loop testing comes in various forms: model-in-the-loop (MiL), software-in-the-loop (SiL) and hardware-in-the-loop (HiL).

MiL testing is performed at the early phases of system development: neither the software, nor the hardware components exist yet, and the tested artifact is a model of the system.

In SiL testing, the actual software is considered. Re-targeting occurs when the software is compiled for a different hardware than the target one (e.g., using a desktop compiler). Re-hosting is preferred for better representativeness: the binary code is the same as the one in the actual system, and it runs on an emulator of the target hardware.

Finally, HiL testing uses the actual software running on the target hardware.

For complex systems, the MiL / SiL / HiL classification might be too schematic. Hybrid forms of in-the-loop testing can be considered, where the tested artifact includes system components having different levels of development. For example, one component is included as a model (MiL), while another one is finalized (HiL). Integrating components with such different levels may however raise difficult interconnection and timing issues.

Methods for evaluation of reliability indicators:

To determine the reliability indicators, two methods are used: non-parametric when you have an unknown type of the law of distribution of operating time before failure, which includes a direct assessment of the reliability indicators from selected data; parametric when you have a known type of the distribution law.

Formulas for determining reliability indicators for parametric and non-parametric methods are given in Table 1.

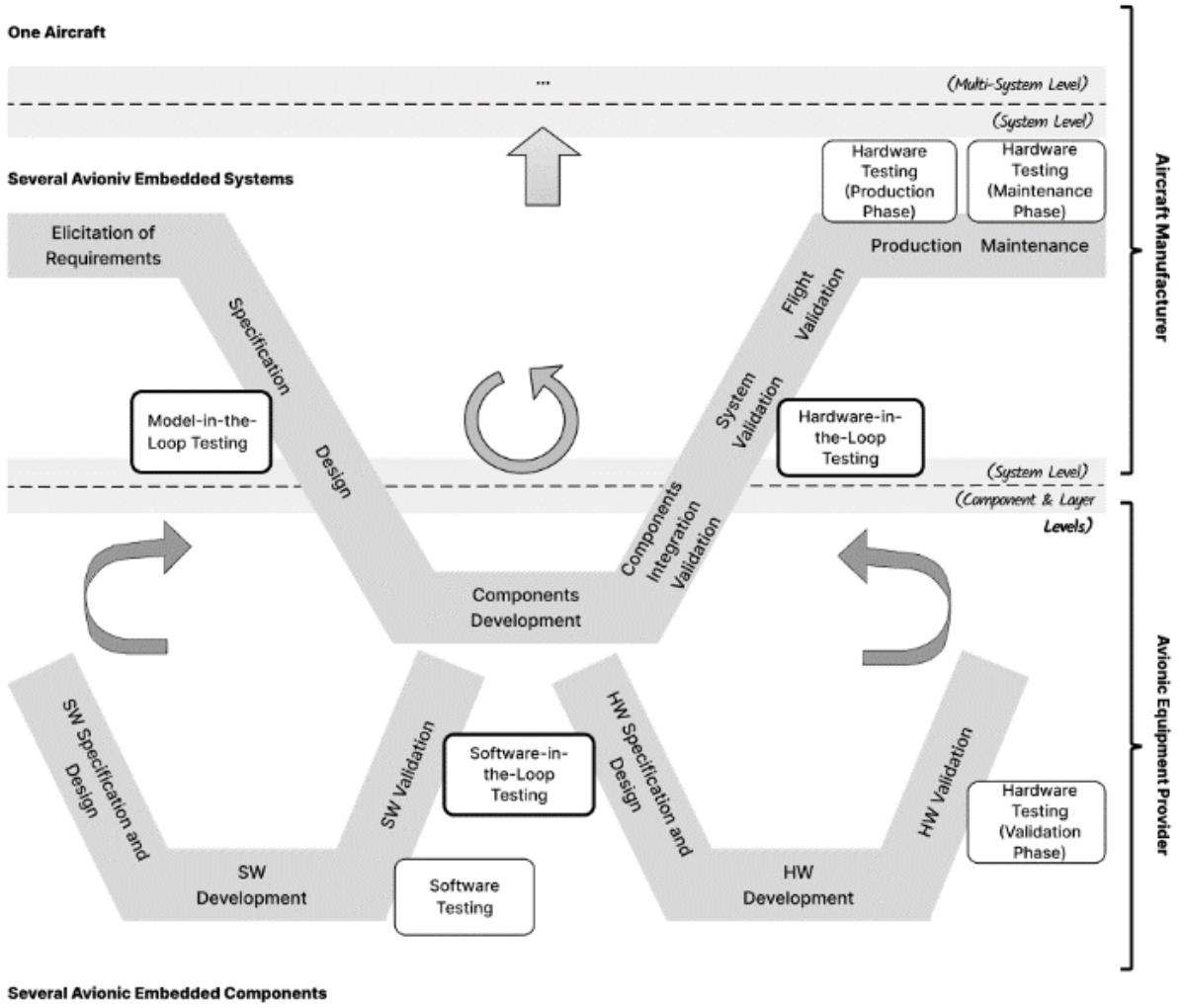

Fig. 1. The life-cycle of an avionic embedded system and the existing types of testing activities

Point and integral assessments of reliability indicators are performed in accordance with GOSTs. In the parametric evaluation of reliability indicators, processing of results of operational observations consists in selection of fault model (observation plan) and type of distribution function (exponential, Weibull, normal, logarithmic normal, gamma, etc.), evaluation of distribution law parameters, verification of acceptance of empirical distribution with its mathematical model (Pearson, Kolmogorov criteria) and evaluation of reliability indicators by parameters of the most adequate distribution law. Formulas for determining confidence limits of reliability indicators and variance of point estimates of reliability indicators and parameters for different distributions are used to evaluate the accuracy of the description. Non-parametric estimates of the main reliability indicators are made depending on the fault model directly according to the data of operational observations.

Table 1. Main indicators of reliability

Reliability indicator		Formulas for calculating reliability indicators	
Name	Designation	Parametric method	Nonparametric method
Probability of uptime	$P(t)$	$\int_t^{\infty} f(t)dt$	$\frac{N-n(t)}{N} = N_{ui}/N$
Operating time before failures	$T_{average}$	$\int_0^{\infty} tf(t)dt$	$1/N\sum_{i=1}^{N} t_i[NUN]$
Failure rate	$\lambda(t)$	$\frac{f(t)}{P(t)}$	$\frac{n(t+\Delta t)-n(t)}{\Delta t[N-n(t)]} = \frac{\Delta n_j}{N_{uj}\Delta t_j}$
Distribution density	$f(t)$	$f(t) = -\frac{1}{P(t)}\frac{dP(t)}{dt}$	$\frac{n(t+\Delta t)-n(t)}{N\times\Delta t} = \frac{\Delta n_j}{N\times\Delta t_j}$
Mean time between failures	T_f	$\int_0^{t} P(t)dt$	$1/n_0\sum_{i=1}^{N} t_i$
Fault flow parameter	$\omega(t)$	$\lim_{\Delta t\to 0}\frac{M_y(t,t+\Delta t)}{\Delta t}$	$\frac{\sum_{i=1}^{N} r_i(t+\Delta t)-\sum_{i=1}^{N} r_i(t)}{N\times\Delta t} = \Delta n_j/t_{sum\,j}$
y-percentage resource	$T_{r.y}$	$1-\int_0^{T_r} f(t)dt = y/100$	$\frac{n(t(i))+1}{N} \le 1-y/100$

$n(t)$ – number of failures during time t; N – number of observed articles; N_{uj} – number of engines operable to the j interval, Δn_j – number of failures in the j interval of Δt_j operating time, n_0 – number of failures that occurred during the period of observing; t_i – operating time of the 1st article; $r_i(t)$ – number of failures of the "i" object to operating time t; t_{sumj} – total operating time of engines in "j" interval; NUN – observation plan according to GOST (N – number of products under supervision, U – non-recoverable and non-replaceable failed products).

There are several different methods for evaluating the parameters of distribution laws, of which the best known are moment, quantile, and maximum likelihood methods. Each of these methods can be effectively applied only in a particular case, using a reasonable compromise between the difficulty of obtaining an estimate and its accuracy.

Consider obtaining evaluation of parameters $\varepsilon_i, i = \overline{1,s}$ of the distribution law $F(\theta,\varepsilon)$ using the listed methods.

According to the moment method, in the presence of theoretical moment characteristics exist, unknown distribution parameters can be expressed through s empirical moments. This approach leads to a system of equations

$$\left\{\int \theta^i f(\theta,\varepsilon_1,\varepsilon_2,\dots,\varepsilon_s)d\theta = m_i^*, i = \overline{1,s} \quad (1)$$

where $f(\theta,\varepsilon)$ is the distribution density of the random variable θ; m_i^* – are empirical initial moments.

Getting the estimation of distribution parameters by the modified quantile method is obtained by minimizing the functional

$$\Phi\big(F(\theta,\varepsilon) - F^*(\theta)\big) \to \min, \quad (2)$$

where $\Phi(\cdot)$ is the objective function; $F^*(\theta)$ is the empirical distribution law.

This operation is carried out by methods of approximating or minimizing the functions of many variables.

The modified quantile method is usually applied if the distribution function is expressed in elementary functions. You can also use this method to find estimates of distribution parameters that do not have initial moments.

The most efficient evaluations are given by the maximum likelihood method, according to which parameters are determined from the condition of the maximum logarithm of the likelihood function

$$\mathrm{LnL}(\vec{\varepsilon}) \to \min, \qquad (3)$$

or

$$\left\{\frac{\mathrm{d}}{\mathrm{d}\vec{\varepsilon}}\theta^{\mathrm{i}}\mathrm{f}(\theta, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{\mathrm{s}})\mathrm{d}\theta = 0\,, \mathrm{i} = \overline{1, \mathrm{s}}, \right. \quad (4)$$

where $\mathrm{L}(\vec{\varepsilon}) = \mathrm{f}(\theta_1, \theta_2, \dots, \theta_{\mathrm{r}}; \vec{\varepsilon}\,) = \prod_{\mathrm{j}=1}^{\mathrm{r}} \mathrm{f}\left(\theta_{\mathrm{j}}, \vec{\varepsilon}\right)$ is the likelihood function, it is also the mutual distribution density of independent random variables $\theta_{\mathrm{j}}, \mathrm{j} = \overline{1, \mathrm{r}}$.

Tables 1, 2 give the expressions for evaluating the parameters of distributing laws, most commonly used in reliability theory and their mean square errors.

Statistical methods of testing hypotheses using the Pearson's consent criteria, Kolmogorov or others are used to select the appropriate distribution [1-4].

In Pearson's criterion, the measure of divergence is taken as the value $x^2{}_0$, the experimental value of which is determined by the formula

$$x_0^2 = \sum_{\mathrm{i}=0}^{\mathrm{k}} \frac{(\mathrm{R_i} - \mathrm{P_i})^2}{\mathrm{rP_i}} \qquad (5)$$

where $\mathrm{P_i} = \mathrm{F}(\theta^*{}_{\mathrm{i}}) - \mathrm{F}(\theta^*{}_{\mathrm{i}-1})$ is the hypothetical probability of falling into the interval $[\theta^*{}_{\mathrm{i}-1}, \theta^*{}_{\mathrm{i}})$; $\mathrm{R_i}$ – the number of statisticians that fell into the interval $[\theta^*{}_{\mathrm{i}-1}, \theta^*{}_{\mathrm{i}})$; k – number of intervals $\mathrm{k} \sim \sqrt[3]{\mathrm{r}}$.

If the measured value by sample is $x^2{}_0$ with the significance level α, then the hypothesis is accepted. With an even value of degrees of freedom $v = \mathrm{k} - \mathrm{s} - 1$, the probability of accepting the hypothesis $\alpha_0 = P(x^2 \geq x^2{}_0)$ can be found by the expression

$$P(x^2 \geq x^2{}_0) = \frac{1}{\Gamma\left(\frac{v}{2} 2^{\frac{\mathrm{v}}{2}}\right)} \int_{\mathrm{x}_0^2}^{\infty} x^{\frac{\mathrm{v}}{2}-1} \mathrm{e}^{-\frac{\mathrm{x}}{2}} \mathrm{d}x = \mathrm{e}^{-\frac{\mathrm{x}}{2}} \sum_{\mathrm{i}=1}^{\frac{\mathrm{v}}{2}-1} \frac{x_0^{\mathrm{i}}}{\mathrm{i}!\, 2^{\mathrm{i}}} \qquad (6)$$

In the Kolmogorov criterion, the probability of accepting the hypothesis is

$$\alpha_0 = \mathrm{P}(\mathrm{D} \geq \mathrm{D}_0) = 1 - \mathrm{k}\left(\sqrt{\mathrm{r}}\mathrm{D}_0\right) \qquad (7)$$

where $\mathrm{D}_0 = \mathrm{m}\left(\mathrm{F}(\theta^*{}_{\mathrm{i}}) - \mathrm{F}^*(\theta^*{}_{\mathrm{i}-1})\right)$; $k(z) = 1 + 2\sum_{j=1}^{\infty}(-1)^i exp(-2i^2z^2)$ Kolmogorov distribution.

Table 2. Evaluating parameters of distribution laws

Distribution law	Analytical expression for the distribution density Method of moments (MM) Maximum likelihood method (MLM)
1	2
Normal	$\frac{1}{\mathrm{S}\sqrt{2\pi}} \exp\left(-\frac{(\mathrm{x}-\mathrm{m})^2}{2S^2}\right) \quad -\infty < \mathrm{x} < \infty;\ \mathrm{S} > 0$
	$\mathrm{m} = \mathrm{m}_1^* = \frac{1}{\mathrm{r}\sum_{\mathrm{i}=1}^{\mathrm{r}} \mathrm{x_i}} \quad \mathrm{S}^2 = \mu_2^* = 1/\mathrm{r}\sum_{\mathrm{i}=1}^{\mathrm{r}} \mathrm{x_i^2} - (\sum_{\mathrm{i}=1}^{\mathrm{r}} \mathrm{x_i})^2$
	$\mathrm{m} = \mathrm{m}_1^*\ \mathrm{S}^2 = \mu_2^*$

Continuation of table 2

1	2
Lognormal	$\frac{1}{xS_L\sqrt{2\pi}}\exp\left(-\frac{(\ln x - m)^2}{2S_L^2}\right)\ 0 < x < \infty;\ S_L > 0$
	$m_L = 2\ln m_1^* - 1/2\ln m_2^*\ S_L^2 = \ln m_2^* - 2\ln m_1^*$
	$m_L = \frac{1}{r}\sum_{i=1}^{r}\ln x_i = M[\ln x] = m_L^*;\ S_L^2 = \frac{1}{r}\sum_{i=1}^{r}(\ln x_i - m_L)^2 = D[\ln x] = S^{*2}$
Gamma distribution	$\frac{1}{\beta^\alpha G(\alpha)}x^{\alpha-1}e^{-x/\beta}\ 0 > x > \infty\ \alpha > 0\ \beta > 0$
	$\alpha = \frac{m_1^{*2}}{\mu_2^*}\ \beta = \frac{\mu_2^*}{m_1^*}$
	β is determined from the solution of the equation $-\ln\beta - \psi\left(\frac{m_1^*}{\beta}\right) + m_L^* = 0;\ \alpha = m_1^*/\beta$ where $\psi(x) = \frac{d}{dz}\ln G(z) = -C - 1/z + \sum_{k=1}^{\infty}\left(\frac{1}{k(z+k)}\right)$
Beta distribution	$\frac{G(a+b)}{G(a)G(b)}x^{\alpha-1}(1-x)^{b-1}\ 0 < x < 1;\ a > 0; b > 0$
	$a = \frac{m_1^{*2} - m_1^* m_2^{*2}}{m_2^* - m_1^{*2}}\ b = a/m_1^* - a$
	a and b are determined from the solution of the system of equations $\begin{cases} \Psi(a+b) - \Psi(a) + 1/r\sum_{i=1}^{r}\ln x_i = 0 \\ \Psi(a-b) - \Psi(b) + 1/r\sum_{i=1}^{r}\ln(1-x_i) = 0 \end{cases}$

Ending of table 2

1	2
Weibull distribution	$C\alpha x^{\alpha-1}e^{-Cx^{\alpha}}\ 0 < x < \infty; C > 0;\ \alpha > 0$
	$\frac{G^2(1+1/\alpha)}{G(1+1/\alpha)} - \frac{m^{*2}}{m_2^*} = 0\ C = \left[\frac{G(1+1/\alpha)}{m_1^*}\right]^{\alpha}$
	α is determined from the solution of the equation $\frac{1}{\alpha} + \frac{1}{r\sum_{i=1}^{r} \ln x_i} - \frac{\alpha 1/r\sum_{i=1}^{r} x_i^{\alpha-1}}{1/r\sum_{i=1}^{r} x_i^{\alpha}} = 0\ C = \frac{1}{1/r\sum_{i=1}^{r} x_i^{\alpha}}$
Exponential distribution	$\lambda e^{-\lambda x}\ 0 < x < \infty;\ \lambda > 0$
	$\lambda = 1/m_1^*$
	$\lambda = \frac{1}{1/r\sum_{i=1}^{r} x_i} = 1/m_1^*$

The GOST provides methods for evaluating reliability indicators with a small number of observations using additional information. Evaluation of reliability indicators of both the product as a whole and its components is carried out by combining experimental information obtained as a result of operational observations or tests and additional information taken from the operation of analog products, analysis of reliability during design and other sources.

Stakeholders:

Historically, the aircraft manufacturer was in charge of all the integration activity for avionic embedded systems. It received components from the aircraft equipment provider. Then it integrated these into systems, until complete multi- system integration within the aircraft.

Nowadays, there is a shift of activity from the aircraft manufacturer towards the equipment providers, as the latter are asked to participate in the first integration phase. Thus, the aircraft manufacturer would now directly receive an integrated avionic embedded system: the equipment providers are becoming system providers. When looking at Figure 1, the horizontal line delimiting the intervention of the providers has a tendency to move upward.

The aircraft manufacturer historically has the needed expertise for setting up the in-the-loop testing activity. This activity, now becoming the responsibility of an avionic system provider, opens an opportunity for collaboration between the two. A new type of interaction emerges, comprising the exchange of information on the tests that were executed by the system provider and the aircraft manufacturer [10]. The exchange could concern test specifications, test procedures implementing the test specifications, or test data traces monitored during the execution of the procedures.

Naturally, each actor has its own internal tools and test platforms that it uses for testing. Inherent incompatibilities between them severely limit the exchanges that can be done. In practice, a test cannot be easily ported from one environment to the other.

The difficulties encountered during these new types of interactions between the different stakeholders have motivated our analysis.

The Interfaces of the System under Test:

In the field of avionics, the interfaces of an avionic embedded system are more or less formally presented in an Interface Control Document (ICD). This name is generic and does not define a standard. Each enterprise is

free to define its own ICD format, or even different formats for different aircraft programs. Whatever the specific format, the document contains information on the interfaces at several hierarchical levels (Figure 2), similar to those present in the OSI model.

At the lowest level, that of physical connections, the connectors of the system under test are presented and given unique identifiers. The pins of each connector are presented as well. Afterwards, the buses and lines that pass through the physical connections are indicated. At a low logical level, the messages are mentioned. Finally, at the highest logical level, the application parameters and signals are described. These represent the data used and produced by the embedded software. A signal corresponds to an instance of an application parameter encoded on a specific bus. For example, the aircraft speed can be sent to two neighbors, using respectively an AFDX connection for the first and an ARINC 429 for the second.

Several types of system network elements are used in the field of avionics for the communication between components, such as the following communication buses:

- Discrete,
- Analog,
- AFDX (Avionics Full-Duplex Switched Ethernet),
- ARINC 429 (Aeronautical Radio, Incorporated),
- MIL-STD-1553B,
- …

For example, let us assume that an avionic embedded component possesses on its interface a connector with a pin conforming to the ARINC 429 standard. This pin is used, naturally, for an ARINC 429 bus. In turn, the ARINC 429 bus communicates several ARINC 429 labels, where each label determines the set of application parameters that constitute the payload of the message. One of these parameters could be the speed of the aircraft. Figure 2 shows what a corresponding ICD would look like.

As mentioned before, the information is organized in a hierarchical manner inside the ICD. There is a tree structure with connectors at the top and application parameters at the bottom. Because such parameters are functionally meaningful to avionics engineers, they are often called engineer variables. We will refer to them as such in the rest of this paper.

The ICD can contain additional information to that presented in the example, like the data type of the engineer variable, its maximum and minimum values, the encoding that was used, or its value refresh rate. As many in-house formats of ICD exist, the supplied information at the various levels can be more or less detailed. In this paper, we assume that the available information is sufficient for a target perimeter of tests.

In a system, several instances of a same engineer variable can be present. For example, such is the case when a component produces an engineer variable that is consumed by several neighboring components. Note that the corresponding interfaces can be of different types. Also, the component producing the parameter may be duplicated within the system for redundancy purposes.

Sample of Test Languages

Table 3 gives an overview of the chosen sample of test languages. The sample consists of:

- four proprietary languages from the avionic domain, which shall be named PL2, PL3 and PL4;
- TestML from the automotive domain, and TTCN-3 (Testing and Test Control Notation Version 3) from the networking and telecommunication domain

The four proprietary test languages, from PL1 to PL4, have been chosen because they represent languages currently employed in the avionics industry. The first one represents the offer of Cassidi-an Test & Services on the U-TEST™ Real-Time System integration test plat-form [4]. To the best of our knowledge, no public test language exists that shows all the characteristics exhibited by these four, and as such, their inclusion was deemed necessary. The fact that we cannot disclose some information does not have a strong im-pact on this paper, as our interest is to discuss general concepts and features of

test languages. In the discussion, we will feel free to use examples of pseudo-code. They will not disclose the precise syntax of proprietary languages but suffice to capture the essence of the identified features.

For comparison purposes, the sample also includes two languages out-side the field of avionics.

TestML is issued from a research project in the automotive domain. Its aim was to investigate the design of an exchange language, in the sense shown by Figure 4. The multiplicity of proprie-tary languages yields the need for many language translators, but if a common exchange language is used then the number of required translators is re-duced. TestML is the only language of our sample that is not operationally used in the industry. It is a research product and its connection to proprietary lan-guages is not implemented. However, it represents an attempt to synthesize con-cerns arising from the in-the-loop testing practice, so that its consideration was deemed relevant to us.

Table 3. The chosen sample of test languages

Test language	Industrial domain of use	Types of testing activities	Types of testing sub-activities	Based on existing language	Specification	Compiled /Interpreted	Standardization status
PL_1	Avionics industry	In-the-loop testing	Model / Software / Hardware-in-the-loop	✓ (OOPL:C++)	Use of libraries	Compiled	-
PL_2			Model-in-the-loop	✓ (OOPL)	Modification of the grammar / Use of libraries	Interpreted	
PL_3			Hardware-in-the-loop	✓ (HSPL)	Use of libraries		
PL_4			Software / Hardware-in-the-loop	-	PL_4 grammar		
TestML [7]	Automotive industry		Model / Software/ Hardware-in-the-loop	-	XML Schemas	-	
TTCN-3 [8]	Networking and telecommunications	Distributed systems and communication protocols testing	-	-	TTCN-3 grammar	Compiled	✓ [9]

References

1. *Hayley J., Reynolds R., Lokhande K., Kuffner M., Yenson S.* Human-systems integration and air traffic control. Lincoln laboratory journal. – № 19. – 2012. – P. 34-49.

2. *Konakhovych H., Kozlyuk I., Kovalenko Y.* Specificity of optimization of performance indicators of technical operation and updating of radio electronic systems of aircraft. System research and information technologies. – 2020. – P. 41-54.

3. *Kovalenko Y.* A programmable logic controller (PLC). Programming language structural analysis. Advances in Intelligent Systems and Computing. – 2017. – P. 234-242.

4. *Kozlyuk I., Kovalenko Y.* Reliability of computer structures of integrated modular aviation for hardware configurations. System research and information technologies. – 2021. – P. 84-94.

5. *Ghannem A., Hamdi M., Kessentini M., Ammar H.* Search based requirements traceability recovery: A multi-objective approach. Proc. IEEE Congress on Evolutionary Computation (CEC). – 2017. – P. 1183–1190.

6. *Z. Jiang, T. Zhao, S. Wang, H. Ju*. New model-based analysis method with multiple constraints for integrated modular avionics dynamic reconfiguration process. – 2020. – 574 p.

7. *Montano G., McDermid J.* Human involvement in dynamic reconfiguration of integrated modular avionics. 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, 2008. – 2008. – P. 4.A.2-1-4.A.2-13.

8. ARINC Specification 653. Avionics application software standard interface. – 2018. [Electronic resource]. – Access point: https://www.sae.org/standards/content/arinc653p3a-1/

9. Committee, AE ARINC 664 Aircraft Data Networks, Part7: Avionics Full Duplex Switched Ethernet (AFDX) Network. Technical Report. – 2005. – 150 p.

10. *Kovalenko Y., Kozlyuk I.* Implementation of the integrated modular avionics application development complex according to the arinc653 standard. The Bulletin of Zaporizhzha National University: Physical and mathematical Sciences. – 2020. – P. 28-36.

Kovalenko Yu.B., Kudrenko S.O.

METHODOLOGY FOR TESTING LANGUAGES FOR EMBEDDED AVIONICS SYSTEMS

In this article, we have analyzed six test languages. Four proprietary languages have been identified that are currently used in avionics for cyclic testing of embedded avionics systems at different levels of integration and maturity of the system under test. We use the Eclipse Modeling Framework with the Ecore specialized modeling language to formalize various concepts of interest. This will allow us to access a number of existing tools to create custom editors, validators, and code generators. Test engineers will have a rich environment to define their own test models based on the meta-model. We propose to abstract from existing proprietary implementation solutions and work at a common design level. For this, mature model design methods exist and can be used. The proposed approach is to share high-level test specifications and automatically maintain the entire code design and production chain.

Keywords: *information support, programming languages, meta-model, integrated modular avionics.*

Коваленко Ю.В., Кудренко С.О.

МЕТОДОЛОГІЯ ДЛЯ ТЕСТУВАННЯ МОВ ДЛЯ ВБУДОВАНИХ СИСТЕМ АВІОНІКИ

У цій статті ми проаналізували шість тестових мов. Було визначено чотири власні мови, які зараз використовуються в авіоніці для циклічного тестування вбудованих систем авіоніки на різних рівнях інтеграції та зрілості системи, що тестується. Ми використовуємо Eclipse Modeling Framework зі спеціалізованою мовою моделювання Ecore для формалізації різних цікавих концепцій. Це дозволить нам отримати доступ до ряду існуючих інструментів для створення спеціальних редакторів, валідаторів і генераторів коду. Інженери-випробувачі матимуть багате середовище для визначення власних тестових моделей на основі метамоделі. Ми пропонуємо абстрагуватися від існуючих пропрієтарних рішень реалізації та працювати на загальному рівні проектування. Для цього існують і можуть бути використані методи розробки зрілих моделей. Запропонований підхід полягає в тому, щоб надати спільний доступ до специфікацій тестування високого рівня та автоматично підтримувати весь ланцюжок розробки та виробництва коду.

Ключові слова*: інформаційне забезпечення, мови програмування, метамодель, інтегрована модульна авіоніка.*